Kryptographie

Grundlagen

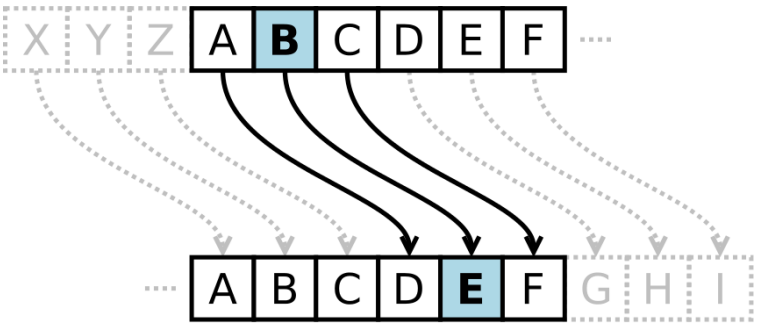
Prof. K. Dohmen
Hochschule Mittweida

18. September 2019

Inhalt

Symmetrische Verfahren

am Beispiel der Caesar-Verschlüsselung



Klar: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Geheim: DEFGHIJKLMNOPQRSTUVWXYZABC

Beispiele

1. CAESAR→ FDHVDU
2. ZAHL → ?

Mathematische Formulierung (1)

Caesar-Verschlüsselung

Wir ordnen den Zeichen des Alphabets Nummern zu:

Buchstabe:	A	B	C	D	E	F	...	Z
Nummer:	0	1	2	3	4	5	...	25

Codierung des Wortes CAESAR durch Zahlenfolge:

$$2, 0, 4, 18, 0, 17.$$

Verschlüsselung durch Verschiebung um $k = 3$ Zeichen:

$$5, 3, 7, 21, 3, 20.$$

Dies entspricht dem Wort FDHVDU.

Mathematische Formulierung (2)

Caesar-Verschlüsselung

Es bezeichne $x \bmod 26$ den Rest, der bei der Division von x durch 26 entsteht, z.B. $38 \bmod 26 = 12$.

Verschlüsselung mit Schlüssel k:

$$c \mapsto (c + k) \bmod 26$$

Entschlüsselung mit Schlüssel k:

$$c \mapsto (c - k) \bmod 26$$

Nachteile der Caesar-Verschlüsselung:

- Je zwei Teilnehmer benötigen einen Schlüssel.
- Ein Angreifer kann den Schlüssel leicht erraten.
- Grundsätzlich sind Substitutionschiffren anfällig gegen Häufigkeitsanalysen.

Ganzzahlige Division mit Rest

Satz und Definition

Seien $a, b \in \mathbb{Z}$ mit $b \neq 0$. Dann gibt es eindeutig bestimmte ganze Zahlen $q, r \in \mathbb{Z}$ (genannt Quotient und Rest) mit

1. $a = qb + r$ und
2. $0 \leq r < |b|$.

Der eindeutig bestimmte Rest wird mit $a \bmod b$ bezeichnet:

$$a \quad = \quad \lfloor a/b \rfloor \, b \quad + \quad (a \bmod b).$$

Beispiele

- $17 \bmod 5 = 2$, da $17 = 3 \cdot 5 + 2$;
- $-6 \bmod 5 = 4$, da $-6 = (-2) \cdot 5 + 4$;
- $34 \bmod 5 = 4$, da $34 = 6 \cdot 5 + 4$.

Kongruenzrelationen und Restklassen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Definition

Sei $n \in \mathbb{N}$. Für $a, b \in \mathbb{Z}$ definieren wir

$$a \equiv b \pmod{n} \quad :\Longleftrightarrow \quad a \bmod n = b \bmod n\,.$$

Satz und Definition

$\equiv$ ist eine Äquivalenzrelation auf $\mathbb{Z}$ für jedes $n \in \mathbb{N}$. Die Äquivalenzklassen werden auch Restklassen genannt.

Eigenschaften der Kongruenz

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Satz

Seien $n \in \mathbb{N}$, $a, b \in \mathbb{Z}$. Folgende Aussagen sind äquivalent:

1. $[a] = [b]$,
2. $a \equiv b \pmod{n}$,
3. $a \bmod n = b \bmod n$,
4. $n \mid a - b$.

Beispiele

- $2 \equiv 0 \pmod{2}$,
- $4 \equiv 1 \pmod{3}$,
- $5 \equiv 9 \pmod{4}$.

Die Menge der Restklassen modulo n

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Satz und Definition

Bezüglich der Kongruenz modulo n gibt es genau n Restklassen, deren Menge wir mit $\mathbb{Z}_n$ bezeichnen:

$$\mathbb{Z}_n = \{[0], [1], \ldots, [n-1]\}.$$

Dabei gilt:

$$[a] = \{b \in \mathbb{Z} \mid a \equiv b \pmod{n}\} \quad \text{für alle } a \in \mathbb{Z}.$$

Beispiele

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Restklassen modulo 3

$\mathbb{Z}_3 = \{[0], [1], [2]\}$ mit

$$\begin{aligned}
[0] &= \{\ldots, -9, -6, -3, 0, 3, 6, 9, \ldots\},\\
[1] &= \{\ldots, -8, -5, -2, 1, 4, 7, 10, \ldots\},\\
[2] &= \{\ldots, -7, -4, -1, 2, 5, 8, 11, \ldots\}.
\end{aligned}$$

Restklassen modulo 4

$\mathbb{Z}_4 = \{[0], [1], [2], [3]\}$ mit

$$\begin{aligned}
[0] &= \{\ldots, -12, -8, -4, 0, 4, 8, 12, \ldots\},\\
[1] &= \{\ldots, -11, -7, -3, 1, 5, 9, 13, \ldots\},\\
[2] &= \{\ldots, -10, -6, -2, 2, 6, 10, 14, \ldots\},\\
[3] &= \{\ldots, -9, -5, -1, 3, 7, 11, 15, \ldots\}.
\end{aligned}$$

Rechnen mit Restklassen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Definition

Sei $n \in \mathbb{N}$. Für zwei Restklassen $[a], [b] \in \mathbb{Z}_n$ definieren wir

- $[a] \oplus [b] := [a + b]$,
- $[a] \otimes [b] := [a \times b]$.

Beispiel

In $\mathbb{Z}_4 = \{[0], [1], [2], [3]\}$ gilt:

$$[2] \oplus [3] = [2 + 3] = [5] = [1], \quad \text{da } 5 \equiv 1 \pmod 4,$$
$$[2] \otimes [3] = [2 \times 3] = [6] = [2], \quad \text{da } 6 \equiv 2 \pmod 4.$$

Verknüpfungtafeln

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Addition und Multiplikation in $\mathbb{Z}_5$

$\oplus$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[1]	[2]	[3]	[4]
[1]	[1]	[2]	[3]	[4]	[0]
[2]	[2]	[3]	[4]	[0]	[1]
[3]	[3]	[4]	[0]	[1]	[2]
[4]	[4]	[0]	[1]	[2]	[3]

$\otimes$	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]
[2]	[0]	[2]	[4]	[1]	[3]
[3]	[0]	[3]	[1]	[4]	[2]
[4]	[0]	[4]	[3]	[2]	[1]

Integerarithmetik in C++

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

```
#include <iostream>
using namespace std;

int main() {

  unsigned int n = 4294967295;  // n = 2^32 - 1
  cout << n*n << endl;

}
```

Es wird „1“ ausgegeben, da in $\mathbb{Z}_{2^{32}}$ gilt:

$$[2^{32}-1] \otimes [2^{32}-1] = [2^{64} - 2 \times 2^{32} + 1] = [1]$$

Einwegfunktionen (1)

leicht berechenbar, schwer invertierbar

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Definition

Eine Einwegfunktion ist eine Abbildung $f : D \to W$, so dass

1. für $x \in D$ die Berechnung von $f(x)$ schnell möglich ist,
2. für $y \in W$ das Auffinden eines $x \in D$ mit $f(x) = y$ *nicht* schnell möglich ist.

Beispiel: Multiplikation

Das RSA-Verfahren basiert auf der Annahme, dass die Multiplikation von Primzahlen eine Einwegfunktion ist:

$$f(p, q) = pq.$$

Einwegfunktionen (2)

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: RSA-704 (212-stellig)

7403756347956171282804679609742957314259318888923128908493623263897276503402826627689199641962511784399589433050212758537011896809828673317327310893090055250511687706329907239638078671008609696253793465056379635 9

Primfaktorzerlegung:

9091213529597818878440658302600437485892608310328358720428512168960411528640933367824950788367956756806141 × 8143859259110045265727809126284429335877899002167627883200914172429324360133004116702003240828777970252499

(Bai, Thomé, Zimmermann, 2012)

Kryptographische Hashfunktionen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Definition

Eine kryptographische Hashfunktion ist eine Einwegfunktion $f : D \to W$, wobei

- D eine Menge von Zeichenketten beliebiger Länge und
- W eine Menge von Zeichenketten fester Länge ist.

$f(x)$ ist eine nahezu eindeutige, kurze Kennzeichnung von x, ähnlich einem Fingerabdruck.

Beispiele

```
kd@xps13:~$ echo "Blockchain Herbstschule" | md5sum
8f6c445aa6f12fde9e6632db4dc852ff
kd@xps13:~$ echo "Blockchain-Herbstschule" | md5sum
d52a5e18090dcacef059c246df8f726a
```

Anforderungen an Hashfunktionen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Resistenz gegen Urbild-Angriffe

Pre-Image-Attack: Zu einem gegebenen Hashwert $y \in W$ wird eine Eingabe $x \in D$ erzeugt mit $f(x) = y$.

2nd Pre-Image-Attack: Zu einer gegebenen Eingabe $x \in D$ wird eine Eingabe $x' \neq x$ erzeugt mit $f(x) = f(x')$.

Resistenz gegen Kollisionsangriffe

Kollisionsangriff: Der Angreifer erzeugt zwei verschiedene Eingaben $x, x' \in D$ mit $f(x) = f(x')$.

In Bitcoin verwendete Hashfunktionen

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

SHA-256

- Aus der Secure-Hash-Algorithm-2-Familie (NSA, 2001)
- erzeugt einen 256-Bit-Hash (64 Hexadezimalstellen)
- Verwendung beim Bitcoin-Mining (Proof-of-Work)

RIPEMD-160

- RIPE Message Digest (Dobbertin, Bosselaers, Preneel, 1996)
- erzeugt einen 160-Bit-Hash
- Verwendung: Bitcoin-Adressgenerierung

Hashpuzzles als Arbeitsnachweise

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Beispiel: Hello, world!

Bestimme $Nonce \in \mathbb{N}_0$ so, dass

$$\text{SHA256}(\text{Hello, world!}Nonce) \leq \text{000f}).$$

```
"Hello, world!0" => 1312af178c253f84028d480a6adc1e25e81caa...
"Hello, world!1" => e9afc424b79e4f6ab42d99c81156d3a17228d6...
"Hello, world!2" => ae37343a357a8297591625e7134cbea22f5928...
...
"Hello, world!4249" => c004190b822f1669cac8dc37e761cb73652...
"Hello, world!4250" => 0000c3af42fc31103f1fdc0151fa747ff87...
```

Puzzle-Eigenschaft

Die Hashfunktion H sollte so beschaffen sein, dass für jede Eingabe x eine Lösung y des Suchproblems $H(xy) \in P$ o.ä. nur mit hohem Rechenaufwand gefunden werden kann.

RSA und El Gamal

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Aus der englischsprachigen Vorlesung Foundations of Modern Cryptography:

RSA

- Initialization phase
- Encryption and decryption
- Correctness of RSA
- Fast exponentiation
- Common modulus attack

El Gamal

- Encryption scheme
- El Gamal signatures
- Discrete logarithms

RSA Initialization phase

Rivest, Shamir, Adleman (1978)

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

- Initialization:
  1. Bob chooses two big primes p and q.
  2. He computes $n = pq$ and $\phi(n) = (p-1)(q-1)$.
  3. He deletes p and q.
- Public key:
  4. Bob chooses $e \in \mathbb{N}$ satisfying $1 < e < \phi(n)$ and
  $$\gcd(e, \phi(n)) = 1.$$
  5. He publicises (n, e).
- Private key:
  6. Bob computes $d \in \mathbb{N}$ satisfying $1 < d < \phi(n)$ and
  $$ed = 1 \pmod{\phi(n)}.$$
  7. He deletes $\phi(n)$ and keeps (n, d) secret.

Encryption and decryption

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Encryption:

- Alice fetches Bob's public key (n, e).
- She encodes her message as $m \in \mathbb{Z}_n$.
- She computes $c = m^e$ in $\mathbb{Z}_n$.
- She sends c to Bob.

Decryption:

- Bob looks up his own private key (n, d).
- He computes $m = c^d$ in $\mathbb{Z}_n$.

Conversion between strings and numbers

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

In Sage

```
symbols = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ\
abcdefghijklmnopqrstuvwxyz+/=\n:;,.!?()@- "

def str2num(s):
    digits = [symbols.find(c) for c in s]
    return ZZ(digits,len(symbols))

def num2str(n):
    s = ''.join([symbols[digit] for digit in
                 n.digits(len(symbols))])
    return s
```


Initialization by example

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Example

Bob chooses two big primes and computes

$$n = 43 \cdot 59 = 2537,$$
$$\phi(n) = (43-1)(59-1) = 42 \cdot 58 = 2436.$$

He then chooses $1 < e < \phi(n)$, which is coprime with $\phi(n)$, e.g., $e = 13$, and computes its inverse in $\mathbb{Z}_{2436}$:

$$13 \cdot d = 1 \pmod{2436}.$$

The extended Euclidean algorithm gives $d = 937$.

- Public key: $(2537, 13)$
- Private key: $(2537, 937)$

Encryption by example

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Example (Cont'd)

Consider the plaintext

PUBLICKEYCRYPTOGRAPHYX.

By replacing each letter by its position $0, \ldots, 25$ in the alphabet, we obtain

$$1520\,0111\,0802\,1004\,2402\,1724\,1519\,1406\,1700\,1507\,2423\,.$$

For $(n, e) = (2537, 13)$ the encrypted message is

$$0095\,1648\,1410\,1299\,0811\,2333\,2132\,0370\,1185\,1457\,1084\,.$$

For example, $1520^{13} = 95$ in $\mathbb{Z}_{2537}$.

RSA Correctness Theorem

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Theorem

Let (n, e) a public RSA key and (n, d) the corresponding private RSA key. Then

$$(m^e)^d = m \quad \textit{for all } m \in \mathbb{Z}_n.$$

The proof is based on Fermat's Little Theorem:

Fermat's Little Theorem (1640)

For any prime p and any $a \in \mathbb{Z}$ where $a \neq 0 \pmod{p}$,

$$a^{p-1} = 1 \pmod{p}$$

Fast exponentiation

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Consider the arithmetic operations performed to compute

$$a^{23} = a\left(a\left(a\left(a^2\right)^2\right)^2\right)^2 = a\left(a\left(a\left(\left(a\cdot 1^2\right)^2\right)^2\right)^2\right)^2$$

Using the abbreviations

- 0 for squaring,
- 1 for squaring and multiplying by a,

these operations can be described by 10111, which is the binary representation of 23:

$$23 = \underline{1}\cdot 2^4 + \underline{0}\cdot 2^3 + \underline{1}\cdot 2^2 + \underline{1}\cdot 2^1 + \underline{1}\cdot 2^0 .$$

Fast exponentiation (cont'd)

Iterative implementation in Python

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

```
def pot(a,n):
    if n == 0: return 1
    n = bin(n)[2:]
    result = 1
    for i in xrange(len(n)):
        result *= result
        if n[i] is '1':
            result *= a
    return result
```

Common modulus attack

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

- A group of people use the same modulus $n = pq$, but with different coprime exponents e_1 and e_2.
- A message is encrypted twice using (n, e_1) and (n, e_2):

$$c_1 = m^{e_1} \pmod{n},$$
$$c_2 = m^{e_2} \pmod{n}.$$

- Eve, who knows the public keys (n, e_1) and (n, e_2), intercepts c_1 and c_2 and computes $s, t \in \mathbb{Z}$ such that

$$se_1 + te_2 = 1.$$

- Without loss of generality, assume that s is negative. Then, provided c_1 is coprime with n,

$$m = (c_1^{-1})^{-s} c_2^t \pmod{n}.$$

El Gamal encryption

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

The El Gamal encryption requires

- a large prime p,
- a primitive root a of p, that is,

$$\{a^1, \ldots, a^{p-1}\} = \{1, \ldots, p-1\} \pmod{p},$$

- an integer $A < p$,
- an integer $B < p$ such that $B = a^A \pmod{p}$.

Alice's private/public key pair:

$$(p, a, A) \quad \text{resp.} \quad (p, a, B).$$

Encryption and decryption work as follows:

Bob chooses $k < p$ at random and encrypts m by the rule

$$(C_1, C_2) = (a^k \bmod p, B^k m \bmod p).$$

Alice recovers m by computing $m = C_1^{-A} C_2 \bmod p$.

El Gamal signatures (1)

System parameters and key generation

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Suppose:

- Alice wants to sign a message m to Bob.
- Bob is to verify her signature.

System parameters:

- a large prime p,
- a primitive root a of p.

Key generation (by Alice):

Private: an integer $A < p$,

Public: an integer $B < p$ such that $B = a^A \pmod{p}$.

El Gamal signatures (2)

Signing and verifying

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Signing the message. Alice chooces a random $k < p - 1$ which is coprime with $p - 1$. She then computes

$$r = a^k \bmod p,$$
$$s = k^{-1}(m - Ar) \bmod p - 1,$$

and sends m along with the signature (r, s) to Bob.

Verifying the signature. Bob computes

$$v_1 = B^r r^s \bmod p, \quad v_2 = a^m \bmod p,$$

using Alice's public key. If $v_1 = v_2$ he accepts the signature.

El Gamal signatures by example

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Example

Alice chooses as her private key $(p, a, A) = (859, 2, 100)$, so her public key is $(859, 2, B)$ where $B = 2^{100} \bmod 859 = 712$.

Suppose $m = 500$. Alice picks $k = 199$ and confirms that $\gcd(k, p-1) = \gcd(199, 858) = 1$. Then she computes

$$r = a^k = 2^{199} = 67 \pmod{859},$$
$$s = k^{-1}(m - Ar) = 199^{-1}(500 - 100 \times 67) = 400 \pmod{858},$$

and sends $(500, (67, 400))$ to Bob. Bob now computes

$$v_1 = B^r r^s = 712^{67} 67^{400} = 175 \pmod{859},$$
$$v_2 = a^m = 2^{500} = 175 \pmod{859}.$$

Since $v_1 = v_2$, he accepts the signature.

Why it works

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

If the signature is not forged, then

$$\begin{aligned} v_1 &= B^r r^s \pmod{p} \\ &= (a^A)^r (a^k)^s \pmod{p} \\ &= a^{Ar+ks} \pmod{p} \\ &= a^{Ar+m-Ar+d(p-1)} \pmod{p} \\ &= a^m (a^{(p-1)})^d \pmod{p}. \end{aligned}$$

By Fermat's Little Theorem, $a^{p-1} = 1$ and hence,

$$\begin{aligned} v_1 &= a^m \pmod{p} \\ &= v_2 \pmod{p}. \end{aligned}$$

Recall that

$$B = a^A \bmod p.$$

Security of the El Gamal scheme is based on the belief that exponentiation in $\mathbb{Z}_p^*$, that is,

$$x \mapsto a^x \bmod p,$$

is a one-way function. Its inverse is called discrete logarithm.

Note

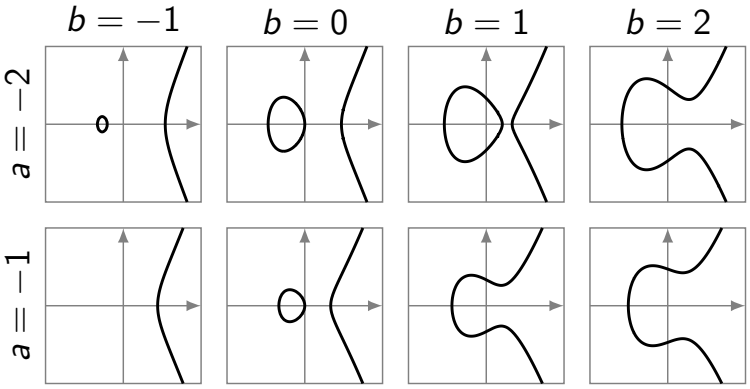
A private El Gamal key is the discrete log of its public key.

Bitcoin makes use of DSA (a variant of El Gamal) and elliptic curve groups.

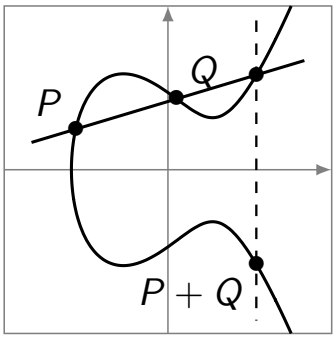
Definition

Elliptische Kurven über einem Körper F sind Punktmengen der Form $\{(x, y) \in F \times F \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$, wobei $4a^3 + 27b^2 \neq 0$ und $\mathcal{O}$ ein unendlich ferner Punkt ist.

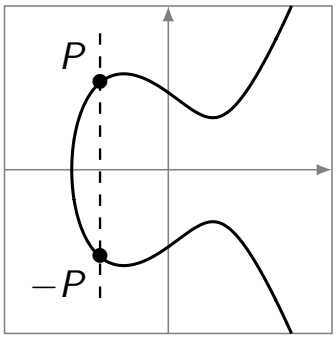
Spezialfall: $K = \mathbb{R}$



Operationen auf elliptischen Kurven



Addition $P + Q$



Inverser Punkt $-P$

Spezialfälle der Addition:

$Q = P$: Man betrachte die Tangente an P statt der Sekante durch P und Q.

$Q = -P$: Die Sekante schneidet die Kurve im unendlichen fernen Punkt $\mathcal{O}$. Daher gilt: $P + (-P) = \mathcal{O}$.

Gruppeneigenschaft und Punktvervielfachung

Satz (Gruppeneigenschaft)

Durch die Addition von Punkten wird eine kommutative Gruppenoperation auf einer elliptischen Kurve definiert.

Definition (Punktvervielfachung)

Für jeden Punkt P einer elliptischen Kurve sei

$$nP := \underbrace{P + P + \cdots + P}_{n \text{ mal}} \quad (n = 1, 2, 3, \dots).$$

Wichtig:

Die Abbildung $n \mapsto nP$ ist für endliche Körper leicht zu berechnen, aber schwer zu invertieren (Einwegfunktion).

Endliche Körper

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Satz

Zu jeder Primzahlpotenz p^n gibt es einen bis auf Isomorphie eindeutig bestimmten endlichen Körper $\mathbb{F}_{p^n}$ mit p^n Elementen. Für $n = 1$ ist dies der Restklassenkörper $\mathbb{Z}_p$.

Addition und Multiplikation in $\mathbb{Z}_5$

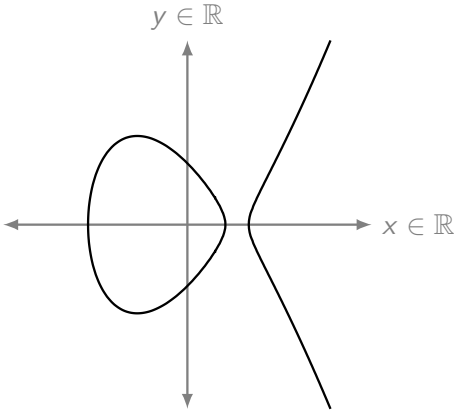
$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

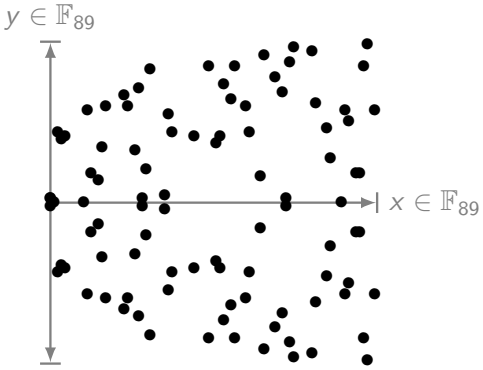
Elliptische Kurven über endlichen Körpern

HOCHSCHULE MITTWEIDA UNIVERSITY OF APPLIED SCIENCES

Spezialfall: $\mathbb{F}_{89}$



$y^2 = x^3 - 2x + 1$ über $\mathbb{R}$



$y^2 = x^3 - 2x + 1$ über $\mathbb{F}_{89}$

Die Bitcoin-Kurve secp256k1

Bitcoin verwendet die NIST-Kurve $y^2 = x^3 + 7$ über $\mathbb{F}_p$, wobei

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

Dabei verwenden alle Bitcoin-Nutzer denselben Generator G als Bezugspunkt und

- wählen individuell und zufällig einen privaten Schlüssel k,
- und bestimmen ihren öffentlichen Schlüssel K durch

$$K = kG = \underbrace{G + G + \ldots G}_{k \text{ times}}.$$

Das Signaturverfahren wird ECDSA genannt (Elliptic Curve Digital Signature Algorithm).

Literatur

A. McAndrew.
Introduction to Cryptography with Open-Source Software.
CRC Press, 2011.

A.M. Antonopoulos.
Mastering Bitcoin, Unlocking Digital Cryptocurrencies.
2nd edition, O'Reilly, Sebastopol, 2017.

K. Dohmen.
Vorlesung "Foundations of Modern Cryptography".
https://www.hs-mittweida.de/dohmen